SesameStreet++

James Rowell

January 11, 2017

There are 10 sorts of people in the world: those who understand binary and those who don't.

Most of us think about "whole numbers" not too differently from the way we learned to count by watching Sesame Street, the difference being that now we can count a little higher. How we've trained ourselves it's automatic to think that the way we write a number or say a number *is* the number.

If I owe you 13 cents and I give you one dime and three pennies then after thanking me profusely for repaying this staggering debt, we'll agree that it's settled with those coins equaling 13 pennies. We identify the symbol "13" very strongly with this particular number - it would be tough to get through life in the modern world without such an automatic process running in our brains. This example highlights what this particular symbol "13" actually means - one dime (1×10) plus three pennies (3×1).

Let's look at the number 13 in some alternative ways - it's the number of months in a year plus one month; what I'm suggesting is that there is no need for the symbol "13" in order to think about this particular number of months. Similarly, 13 is this many apples 🍎🍎🍎🍎🍎🍎🍎🍎🍎🍎🍎🍎🍎; or 13 is the sixth prime number. None of these ways of thinking about the number 13 require that we represent it using the digits 1 and 3 butted up next to each other.

Each number exists independently from any symbol or word that might represent it. Numbers are an idea - perhaps such a strong idea that the universe wouldn't exist without them! Anyway, for our purposes whole numbers exist in some abstract realm - Each number is one whole unit more than the previous number, starting at nothing, that is "zero", and jumping to something, that is "one", then one more, which gets us to "two", then again to "three", etc. Continuing in this way forever... we get them all.

To get back to the idea of what a whole number really is, try to forget about the symbols or words we use and picture a pile of apples. There's zero apples (it's hard to show no-apples), then we introduce an "🍎" to get our very first, and smallest, non-empty pile of apples. Then add another apple to get a pile of "🍎🍎", then "🍎🍎🍎", then "🍎🍎🍎🍎" then some big pile of "🍎🍎🍎🍎🍎...🍎🍎🍎" after we've been adding apples for a while. Each successively bigger pile of apples corresponds with each successive whole number.

We expand this entire set of whole numbers to include their negative-counterparts and call this larger set "integers". We denote the set of integers with this symbol: $\mathbb{Z}$. If we only want to talk about positive integers along with zero, we use this symbol: $\mathbb{Z}_{\geq 0}$.

However, using a "1" followed by a "3" to represent "🍎🍎🍎🍎🍎🍎🍎🍎🍎🍎🍎🍎🍎" is VERY handy. So we use Hindu-Arabic numbers and the positional notation of "base-ten", more commonly known as “decimal”, to represent each specific integer. We slap a “-” on the front if we need to talk about a negative integer.

Base-ten representation of an integer is far superior to ancient Roman numerals for example. Try adding two numbers together in ancient Rome, or worse, multiplying or dividing them. What’s XI times IX? Would you believe me if I told you it’s XCIX? Unless you convert those to Hindu-Arabic numerals to check, you’re just gonna have to trust me. Truth is - I don’t know how to multiply using Roman numerals - nor did most Romans! Not only that, but I’ll bet that most kids who graduated from Sesame Street can count higher than any Roman could - as the Roman system only effectively allowed counting up to 4999.

Using base-ten for us is automatic, we barely think about it when we’re adding numbers or multiplying them - but it’s worth looking carefully at how base-ten works - so let’s examine it from the ground up.

It’s useful to have simple symbols to represent each of the integers from one to nine, namely our familiar 1, 2, 3, 4, 5, 6, 7, 8 and 9 which have an interesting history and predate their use in base-ten.

Slightly more modern, but still quite ancient, is the symbol “0” for “zero”, originally meaning “empty”. Zero also predates its use in base-ten but without zero, base-ten wouldn’t be possible.

Base-ten uses the idea of stringing a series of digits together (a digit being one of the numbers 0, 1, 2, ... 9), one after the other to be able to represent any whole number. Let’s look at the first two-digit number, that is, ten, which as you well know looks like this: “10”. This extra digit on the left tells us how many tens we have and the last, or rightmost digit says how many additional units to add to it.

So our very first two-digit number 10 means “one lot of ten - plus zero units”. When we see “11” - we interpret it to mean “one lot of ten - plus one unit”, and “12” is “one lot of ten - plus two units”, etc. Continuing on; “20” - we interpret to mean “two lots of ten, plus zero units”, etc. up to “90” meaning “nine lots of ten, plus zero units”.

Following this line of reasoning since “10” now means the integer ten, then “100” must mean “ten lots of ten, plus zero units”- which is exactly what it means. We have a special word for this number we call it “one hundred” or “one lot of a hundred, plus zero lots of tens, plus zero units”. Similarly “200” means “two lots of a hundred, plus zero lots of ten, plus zero units”, etc.

We can keep going by one-hundred until we similarly get to “1000” or “ten lots of a hundred, plus zero lots of ten, plus zero units” otherwise known as “a thousand” or more specifically “one lot of a thousand, plus zero lots of a hundred, plus zero lots of ten, plus zero units”.

It get’s a little tedious to be so specific when reading out a number so our language has developed quite a few verbal shortcuts. Furthermore it doesn’t take long before we run out of fancy names for these “powers of ten” like, million, billion, trillion, zillion etc. So let’s introduce some nice clean mathematical notation to describe these powers of ten and let’s forget the fancy words.

$$
\begin{aligned}
100 &= 10 \times 10 = 10^2, \\
1000 &= 10 \times 10 \times 10 = 10^3, \\
10000 &= 10 \times 10 \times 10 \times 10 = 10^4, \\
&\dots \\
\underbrace{10\dots000}_{\text{k zeros}} &= \underbrace{10 \times 10 \times 10 \times 10 \times \dots \times 10}_{\text{k 10s}} = 10^k
\end{aligned}
$$

The 10^k means there are k 10's multiplied together - also written as a 1 followed by k zeros. The above list shows the cases for $k = 2$, 3 and 4. Using the k like that is just a way to show that we can pick ANY whole number, i.e., there is no limit on how big k can be.

The notation of 10^k is very handy, in fact it extends to the case when $k = 0$ and $k = 1$.

So 10^1 means that there is only one 10 multiplied together, or one "0" following the "1", in other words just the number 10 itself.

How about when $k = 0$. Examining the pattern of how the power k relates to how many zeros follow the "1" (eg, $10^1 = 10$, $10^2 = 100$, $10^3 = 1000$, etc.) then it makes sense that $10^0 = 1$, i.e., no zeros follow the "1", which is exactly right. Actually any number raised to the 0^{th} power is 1, but we'll leave that explanation for another time.

Let's look at an example. Reading the number 46307 out according to our technique we can see that it's "four lots of ten-thousand, plus six lots of a thousand, plus three lots of a hundred, plus zero lots of ten, plus seven units":

$$
\begin{array}{rrcr}
 & 4 & \times & 10000 \\
+ & 6 & \times & 1000 \\
+ & 3 & \times & 100 \\
+ & 0 & \times & 10 \\
+ & 7 & \times & 1
\end{array}
\quad = \quad
\begin{array}{rr}
 & 40000 \\
+ & 6000 \\
+ & 300 \\
+ & 00 \\
+ & 7 \\
\hline
= & 46307
\end{array}
$$

Written in terms of powers of ten: $46307 = 4 \times 10^4 + 6 \times 10^3 + 3 \times 10^2 + 0 \times 10^1 + 7 \times 10^0$.

You can think of each digit as being a little dial that controls how many lots of its corresponding power of ten will contribute to the value of the integer.

Claim: Given that we can use as high as power of ten as we like and we can string together as LONG A LIST of digits as pleases us, that means that we can create ANY INTEGER WE WANT no matter how big it is.

That's a pretty tall claim. How do we know that we can create ALL the nonnegative integers with this scheme? For example, how do we know that we didn't miss one? Or how do we know that some string of digits doesn't represent two different integers? I know it seems silly to ask that, but attention to these kinds of details is what is referred to as "rigor" in Mathematics - it's necessary so we don't end up fooling ourselves or spouting bullshit - or if we are full of it then it's easy for other Mathematicians to call us on our nonsense.

We're going to jump into the deep end and make our claim in a careful mathematical way. Such a careful statement is called a theorem - theorems require proof, which we're going to do!

Base-Ten Representation Theorem

Let $n, k \in \mathbb{Z}_{\geq 0}$. Then every n can be uniquely expressed as follows:

$$n = d_k 10^k + d_{k-1} 10^{k-1} + \cdots + d_2 10^2 + d_1 10^1 + d_0 10^0$$

for some k such that $0 \leq d_i \leq 9$ where $d_i, i \in \mathbb{Z}$ and $0 \leq i \leq k$.

Furthermore $d_k \neq 0$ except when $n = 0$.

Definition: n is represented in base-ten as $d_k d_{k-1} \ldots d_2 d_1 d_0$

A difficulty many folks have with math is the notation - it's a kind of a language unto itself - like a computer program is a language. Let's take our theorem statement by statement and turn it into english.

i) "Let $n, k \in \mathbb{Z}_{\geq 0}$"

This means we are going to talk about two distinct numbers that we are labeling n and k. That strange looking $\in$ means "is an element of" (or "is a member of") and is always followed by something that is a "set". We talked above about the symbol $\mathbb{Z}_{\geq 0}$ which we defined as being the set of nonnegative integers. So, in other words, n can be one of 0 or 1 or 2 or 3 or ... any number - no matter how large - and the same goes for k.

This might be what it would sound like to read that line out loud:
"Let n and k be elements of the set of nonnegative Integers."

ii) "Then every n can be uniquely expressed as follows"

What we are about to say applies to ALL nonnegative integers and furthermore the expression is going to be unique for each integer.

iii) "$n = d_k 10^k + d_{k-1} 10^{k-1} + \cdots + d_2 10^2 + d_1 10^1 + d_0 10^0$"

This is the expression in question. It equates n with a series of multiplications of some numbers (the d_i terms where i can be any number from 0 to k) times descending powers of 10, and adds them all together.

It's useful to point out the meaning of our "$d_0, d_1, d_2, \ldots, d_k$" and d_i terms. Mathematical formulas such as this make judicious use of "subscripts" when coming up with names for lists of variables or constants. Subscripts following a letter or symbol, such as $d_0, d_1, d_2, \ldots$ are a handy way to get a list of variable or constant names that are similar looking to each other, and is meant to imply that they each fulfill a similar role to each other. Note here how the value of the subscript on each "d" corresponds to its power of ten, even in the single digit case when the subscript is 0, or the highest power case when the subscript is k. Please also note that it isn't necessary that the subscript indices match with the powers, but it seems helpful!

If we had to read the line out loud it might sound something like this:
"n is equal to... *dee-kay* times ten-to-the-*kay*, plus *dee-kay*-minus-one times ten-to-the-*kay*-minus-one, plus etc. etc., down to... *dee-two* times ten-squared, plus *dee-one* times ten, plus *dee-zero* times one".

iv) "for some k such that $0 \leq d_i \leq 9$ where $d_i, i \in \mathbb{Z}$ and $0 \leq i \leq k$"

The "for some k" means that each integer n has a specific k associated with it.

It then states that those d_i terms are integers, and can ONLY take on the values 0, 1, 2, 3, 4, 5, 6, 7, 8 or 9. Note that our uniqueness claim above means that each integer n has it's own unique list of d's.

It also is very fastidiously pointing out that the little "i" we just introduced in the subscript of the d's is also an integer and can be as small as zero but only as large as our highest power k - whatever k might be. This is very picky stuff - like a computer program spelling things out very precisely so the computer knows exactly what you mean. (That's right - you are the computer).

Sounding it out might sound like this:
"for some *kay* such that zero is less-than-or-equal-to *dee-i* which is less-than-or-equal-to nine, for each *dee-i* and i, which are integers; also i is between zero and k inclusive"

v) "Furthermore $d_k \neq 0 \ldots$"

This is spelling out one more important fastidious detail. We want to make sure that the "most significant d", that is, our d_k that goes along with the highest power 10^k is not 0, in other words it must be one of 1, 2, 3, 4, 5, 6, 7, 8 or 9. This is necessary so that we can get our uniqueness property, otherwise we could say $13 = 013 = 0000013$ which are all the integer 13, so let's outlaw this uninteresting and annoying possibility.

vi) "$\ldots$ except when $n = 0$"

. . . completing that last statement which allows for one exception to the requirement that the "most significant digit" is not allowed to be zero, and that's exactly when the integer n in question IS zero.

vii) "Definition: n is represented in base-ten as $d_k d_{k-1} \ldots d_2 d_1 d_0$"

This is introducing what it means to write the number out in base-ten; that is, we toss out all the extraneous stuff from our expression in (iii) above, and string all the "digits" one after another, from most significant digit d_k on the left down to least significant digit d_0 on the right.

Before we prove our theorem, consider that base-ten is not the only base in use these days. Since the introduction of the EDVAC[1] computer, around 1950, there have been many orders of magnitude more calculations done in base-two (otherwise known as binary) by computers than have EVER been done by people in base-ten for the entirety of human history. (This might even be true if we only count one-day's worth of binary computer calculations - someone needs to check this.)

Binary-computer logic gates (the building blocks of the modern computer) can only take one of two states, that is; "off" or "on". We interpret these two states to represent these two numbers: 0 and 1. By doing so, in the same way that base-ten uses ten numbers 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 for its digits; we can represent integers in base-two with just the digits 0 and 1. How is this possible? Let's find out with an imaginary trip into space.

Consider distant Planet-Nova on which the emergent intelligent species only have nine fingers. They have three hands with three fingers each - anyway, that's why they use base-nine, so they only need the numbers 0, 1, 2, 3, 4, 5, 6, 7 and 8 for their digits. So like we Earthlings do for the

[1] You might be thinking, don't you mean ENIAC which was eariler? Actually no - the ENIAC used base-ten accumulators, not binary!

integer ten, instead of making up a new symbol for nine, they use "10" to represent the integer nine - which for them means "One lot of nine, plus zero units".

Similarly on Planet-Ocho, since they only have eight fingers, then they use base-eight and only use numbers 0, 1, 2, 3, 4, 5, 6 and 7 for their digits. For them "10" means "One lot of eight, plus zero units".

On and on past Planet-Gary-Seven, and Planet-Secks, Planet-Penta, ...

Finally we come upon Planet-Claire (well someone has to come from Planet-Claire, I know she came from there), where the poor blighters only have two fingers so they only use the digits 0 and 1 and base-two, so for them "10" means "one lot of two and zero units". So on Planet-Claire "10" means two. Recall above how we arrived at our 100 in base-ten, being "ten lots of ten, plus zero units" - similarly on Planet-Claire "100" in base-two for them means "Two lots of two plus zero units" in other words, four! What is "11" in base-two? Using our technique to describe the digits we see that it's "One lot of two, plus one unit", in other words three.

Here's how they count on Planet-Claire using base-two:

base-two	base-ten	base-two	base-ten
0	0	(...cont)	
1	1	1101	13
10	2	1110	14
11	3	1111	15
100	4	10000	16
101	5	10001	17
110	6	...	
111	7	11111	31
1000	8	100000	32
1001	9	...	
1010	10	1000000	64
1011	11	10000000	128
1100	12 (cont...)	100000000	256

Note something interesting in the list above - the powers of two, written in base-two, resemble our powers of 10 in base-ten! That is: $1 = 2^0 = 1$, $2 = 2^1 = 10_{(\text{base-2})}$, $4 = 2^2 = 100_{(\text{base-2})}$, $8 = 2^3 = 1000_{(2)}$, $16 = 2^4 = 10000_{(2)}$, $32 = 2^5 = 100000_{(2)}$, $64 = 2^6 = 1000000_{(2)}$, $128 = 2^7 = 10000000_{(2)}$, $256 = 2^8 = 100000000_{(2)}$, ...

Let's look at the binary number 11010 for example. Using our wordy technique to describe the number we can see that it's "One lot of sixteen, plus one lot of eight, plus zero lots of four, plus one lot of two, plus zero units":

$$
\begin{array}{rrcr}
 & 1 & \times & 10000 \\
+ & 1 & \times & 1000 \\
+ & 0 & \times & 100 \\
+ & 1 & \times & 10 \\
+ & 0 & \times & 1
\end{array}
\quad = \quad
\begin{array}{rrl}
 & 10000 & (16) \\
+ & 1000 & (8) \\
+ & 000 & \\
+ & 10 & (2) \\
+ & 0 & \\
\hline
= & 11010 & (26)
\end{array}
$$

Written in terms of powers of two: $11010 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$.

Each digit in base-two can be thought of as a little switch that turns on or off the contribution of its corresponding power of two.

Claim: Given that the inhabitants of Planet-Claire can use as high a power of two as they like, and that they can string together as LONG A LIST of binary-digits as pleases them, that means that they can create ANY INTEGER THEY WANT no matter how big it is.

Sound familiar? Let's restate our theorem for base-ten but rewritten for base-two.

Base-Two Representation Theorem

Let $n, k \in \mathbb{Z}_{\geq 0}$. Then every n can be uniquely expressed as follows:

$$n = d_k 2^k + d_{k-1}2^{k-1} + \cdots + d_2 2^2 + d_1 2^1 + d_0 2^0$$

for some k such that $0 \leq d_i \leq 1$ where $d_i, i \in \mathbb{Z}$ and $0 \leq i \leq k$.

Furthermore $d_k \neq 0$ except when $n = 0$.

Definition: n is represented in base-two as $(d_k d_{k-1} \ldots d_2 d_1 d_0)_2$

Try reading the above out loud in your head, line by line, item by item, like we did above when we "sounded it out" for the base-ten theorem - it's helpful to turn the "math-code" into understandable english and a useful habit to get into when reading mathematical statements.

Before we go on, I want to introduce a little notation to help avoid confusion. How do you know what I'm talking about if I just write "1000"? Do I mean 10^3 or 2^3? If there is any possibility for confusion we write the number like this $(1000)_{10}$ for the base-ten version meaning one-thousand and $(1000)_2$ for the binary version meaning eight.

As is hinted by the habits of our various alien friends above it seems that we can use ANY integer greater than or equal to 2 as a base (base-one doesn't really make sense - think about it for a while). In fact computer graphics artists are known to stumble upon numbers written in hexadecimal (usually relating to specifying a color-channel), which is base-sixteen.

Base-sixteen introduces some new single-character symbols to the usual numbers 1, 2, thru 9, to represent the numbers 10, 11, 12, 13, 14 and 15. Base-sixteen adds the "digits" A, B, C, D, E and F where $\mathrm{A}_{16}=(10)_{10}$, $\mathrm{B}_{16}=(11)_{10}$, $\mathrm{C}_{16}=(12)_{10}$, $\mathrm{D}_{16}=(13)_{10}$, $\mathrm{E}_{16}=(14)_{10}$, $\mathrm{F}_{16}=(15)_{10}$. So if you see this number $(\mathrm{80FB})_{16}$ then I bet at this point (if you take a little time with a calculator and a pad of paper and pencil) then you can figure out that it's $(33019)_{10}$.

Note that if we omit the parentheses and subscript from a number, it means we're talking about it in base-ten - our "default" base. Case in point: the subscripts that we use to denote the base (like the "16" in $(\mathrm{80FB})_{16}$) are written in base-ten!

We really need to get on with proving our two theorems above. But what about proving the "base-nine" version of the theorem for the aliens on Planet-Nova, or the "base-eight" version for the inhabitants of Planet-Ocho?

To cover all bases (pun intended) let's restate our theorem for the general case, call it "base-b", where b is any number greater than or equal to two. If we can prove that theorem, then we'll automatically get all the cases of specific bases for free.

Basis Representation Theorem

Let $n, k, b \in \mathbb{Z}_{\geq 0}$ such that $b \geq 2$. Then every n can be uniquely expressed as follows:

$$n = d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0$$

for some k such that $0 \leq d_i \leq (b-1)$ where $d_i, i \in \mathbb{Z}$ and $0 \leq i \leq k$.

Furthermore $d_k \neq 0$ except when $n = 0$.

Definition: n is represented in base-b as $(d_k d_{k-1} \ldots d_2 d_1 d_0)_b$

As we discussed way up at the top of this essay, we think about positive integers as a process that builds them up one by one. That is, each successive integer is one more than the previous one, starting at 1, then one more taking us to 2, then 3, 4, 5, ... ad infinitum[2]...

This idea of being able to step one after the other, beginning at 1 and going forever is called the "Principle of Mathematical Induction" and is a basic property of the positive integers. This principle is more than just a way to generate the set of integers, it's also a way of thinking about properties of the integers.

Suppose[3] that $P(n)$ means that the property P holds for the number n; where n is a positive integer. Then the principle of mathematical induction states that $P(n)$ is true for ALL positive integers n provided that:

i) $P(1)$ is true

ii) Whenever $P(k)$ is true, $P(k+1)$ is true.

Why would these two conditions show that $P(n)$ is true for all positive integers? Note that condition ii) only asserts the truth of $P(k+1)$ under the assumption that $P(k)$ is true. However if we also know that $P(1)$ is true then condition ii) implies that $P(2)$ is true, which again implies that $P(3)$ is true, which in turn leads to the truth of $P(4)$, etc., over and over for all positive integers.

Some people picture an infinite row of dominoes. Having condition i) (called the "base case") is like being able to knock over the first domino. Then knowing condition ii) is also true is like the fact that any one domino has the ability to knock over the next. Once you've knocked over the first domino, they all fall.

Let's look at a simple example: Perhaps you've heard the story of young Carl Friedrich Gauss as a young boy in the 1780s who was assigned (along with all his classmates) the tedious task of summing the first 100 integers - Presumably to keep them quiet and busy while the teacher corrected some papers. Anyway, young Gauss immediately produced the answer 5050, before most of the boys had summed the first couple of numbers. It wasn't young Gauss's extraordinary computational speed which allowed him to perform this dazzling task, but he had the deeper insight that instead of adding 1 plus 2, then adding 3, then 4, etc. he saw that if you paired 1 with 100, and 2 with 99, and 3 with 98, etc., that each of those pairs added up to 101, furthermore he knew he'd have 50 such pairs, meaning he could state the result in a heartbeat "5050". Gauss is widely regarded as being one of the greatest mathematicians who has ever lived - the young eight-year old was just getting started.

[2] "ad infinitum" means "to infinity", or "continue forever, without limit".

[3] The wording of this definition of "The Principle of Mathematical Induction" is borrowed from "Calculus" by Michael Spivak - an fabulous introductory textbook to Analysis.

Anyway, to generalize Gauss's insight we can write the expression like this:

$$1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2}$$

So let's prove this relationship using the principle of mathematical induction.

Let $n = 1$ for the "base case", then

$$\frac{1(1+1)}{2} = \frac{2}{2} = 1$$

Which is the trivial sum of the first positive integer 1.

Now let's assume the relationship is true for n, and prove that it must also be true for $n+1$:

$$\begin{aligned}
&(1 + 2 + 3 + \ldots + n) + (n+1) \\
&= \frac{n(n+1)}{2} + (n+1) \\
&= \frac{n(n+1)}{2} + \frac{2(n+1)}{2} \\
&= \frac{n^2 + n + 2n + 2}{2} \\
&= \frac{n^2 + 3n + 2}{2} \\
&= \frac{(n+1)(n+2)}{2} \\
&= \frac{(n+1)((n+1)+1)}{2}
\end{aligned}$$

Which proves young Gauss's expression is true for the positive integer $n + 1$ whenever it's true for n - then by the principle of mathematical induction, the expression is true for all positive integers.

We are going to use the principle of mathematical induction to prove the Basis Representation Theorem.

First we will prove that there is such a representation for all integers n (existence proof). Meaning that every integer has a way of being written in the form described by the theorem - especially as relates to the restrictions on the values that the "digits" can take on. After that has been established we will use another technique to prove that each such representation is unique - in other words there aren't two (or more) ways to represent the same integer in base-b.

Existence Proof of the Basis Representation Theorem

Base case:

Since $n = 0$ is a slightly special case in the theorem, then lets look at both $n = 0$ and $n = 1$ to start our base case.

Let $n = 0$.

We can choose $k = 0$ and $d_0 = 0$. (This is the one exception spelled out in the theorem in which the most significant digit of n is allowed to be zero.) Then,

$$n = d_0 b^0 = 0 \times b^0 = 0$$

showing that we have a valid representation for 0 in base-b since our only digit $d_0 = 0 \leq (b - 1)$, for all $b \geq 2$.

Now Let $n = 1$.

In this case, we can choose $k = 0$ and $d_0 = 1$. Then,

$$n = d_0 b^0 = 1 \times b^0 = 1 \times 1 = 1$$

showing that we have a valid representation for 1 in base-b since $d_0 = 1 \leq (b - 1)$, for all $b \geq 2$.

Induction Case:

Assume that n has a valid representation in base-b, that is, n can be written thus:

$$n = d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0$$

with all the appropriate conditions holding for the values of d_i, b and k; and we will prove that $n + 1$ also has a valid representation in base-b.

We're going to break this step into two cases which cover all possibilities.

Case 1) $d_0 \leq (b - 2)$

This case examines when the least significant digit of n is *strictly-less-than* the largest value it can take in base-b. For example, in base-two d_0 can only be zero; In base-five d_0 can be at most three; In base-ten d_0 can be at most eight, etc. This case is quite easy to deal with, so let's quickly dispense with it.

$$\begin{aligned} n &= d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0 \\ &\text{if and only if,} \\ n + 1 &= d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0 + 1 \\ &= d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0 + b^0 \\ &= d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + (d_0 + 1) b^0 \end{aligned}$$

and our by assumption that $d_0 \leq (b - 2)$, then

$$(d_0 + 1) \leq (b - 2) + 1 = (b - 1)$$

showing us that the "least significant digit" of $n+1$, being (d_0+1), is less than or equal to $(b-1)$ which means that (d_0+1) is a valid digit in base-b.

Since all the other d_i terms $(d_1, d_2, \ldots, d_k)$ for $n+1$ are unchanged from their values for n then all the digits of $n+1$ are valid in base-b.

Therefore we've established the truth of "Case 1" for the integer $n+1$.

Case 2) $d_0 = (b-1)$

Now we'll look at the case when the least significant digit of n is equal to the largest value it can take in base-b, that is, $d_0 = (b-1)$. (Note that between "Case 2" here and "Case 1" above, we're covering all the possibilities for what d_0 can be.) For example in base-two $d_0 = 1$; in base-five $d_0 = 4$; in base-ten $d_0 = 9$, etc.

Let $j \in \mathbb{Z}_{\geq 0}$ be the lowest power of b such that $d_j < (b-1)$, meaning we can write n as follows for some j:

$$n = d_k b^k + d_{k-1} b^{k-1} + \cdots + d_j b^j + (b-1)b^{j-1} + \cdots + (b-1)b^1 + (b-1)b^0$$

For example, if $n = 69412999$, then $j = 3$, since 10^3 is the lowest power of 10 such that its digit d_3 is less than 9 (it's 2).

$$\begin{aligned}
n &= d_k b^k + \cdots + d_j b^j + (b-1)b^{j-1} + \cdots + (b-1)b^1 + (b-1)b^0 \\
&= d_k b^k + \cdots + d_j b^j + (b^j - b^{j-1}) + (b^{j-1} - b^{j-2}) + \cdots + (b^2 - b^1) + (b^1 - b^0) \\
&= d_k b^k + \cdots + (d_j b^j + b^j) + (-b^{j-1} + b^{j-1}) + \cdots + (-b^2 + b^2) + (-b^1 + b^1) - b^0 \\
&= d_k b^k + \cdots + (d_j + 1)b^j - b^0 \\
&= d_k b^k + \cdots + (d_j + 1)b^j - 1
\end{aligned}$$

Therefore,

$$\begin{aligned}
n + 1 &= d_k b^k + \cdots + (d_j + 1)b^j - 1 + 1 \\
&= d_k b^k + \cdots + (d_j + 1)b^j \qquad (1)
\end{aligned}$$

Since we picked j such that $d_j < (b-1)$, less restate the inequality as
$d_j \leq (b-2)$ therefore,

$$(d_j + 1) \leq (b-2) + 1 = (b-1)$$

meaning the j^{th} digit of $n+1$ is a valid base-b digit.

All digits $d_k, \ldots, d_{j+1}$ remain unchanged from the base-b representation of n, and all digits $d_{j-1}, \ldots, d_0$ are 0.

Therefore all the digits of the base-b representation of $n+1$ are valid in base-b.

If you've been fastidiously following the conditions on our subscript j above, then you may notice that our proof doesn't leave room for the case that *all* the digits are equal to $(b-1)$ because of how we defined j. For example when $n = 99999$.

Let's attend to this remaining situation.

Suppose $d_i = (b-1)$ for all $0 \leq i \leq k$, then let $d_{k+1} = 0$ and $j = k+1$.

So picking up at equation (1) with our new terms we have:

$$\begin{aligned} n+1 &= (d_j+1)b^j \\ &= (d_{k+1}+1)b^{k+1} \\ &= (0+1)b^{k+1} \\ &= b^{k+1} \end{aligned}$$

Meaning that $n+1$ now has a $(k+1)^{\text{st}}$ digit and it's equal to 1, with all the rest of the digits being 0 - which is a valid representation for $n+1$ in base-b for all $b \geq 2$.

QED[4] - existence proof.

Uniqueness Proof of the Basis Representation Theorem

Suppose n is not unique and that,

$$n = d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b^0$$

also suppose,

$$n = c_k b^k + c_{k-1} b^{k-1} + \cdots + c_2 b^2 + c_1 b^1 + c_0 b^0$$

Let's further suppose that the index j is the first power such that the digits $d_j \neq c_j$ and without any loss of generality, let's assume that $dj > cj$.

Therefore:

$$c_k b^k + c_{k-1} b^{k-1} + \cdots + c_2 b^2 + c_1 b^1 + c_0 b^0 = d_k b^k + d_{k-1} b^{k-1} + \cdots + d_2 b^2 + d_1 b^1 + d_0 b$$

if and only if,

$$0 = (d_k - c_k)b^k + (d_{k-1} - c_{k-1})b^{k-1} + \cdots + (d_j - c_j)b^j$$

[4] "QED" - is often used at the conclusion of a proof to state that it's done - it's an acronym for the latin phrase "quod erat demonstrandum" which means "that which was to be demonstrated". In other words we've proven the existence part of the Basis Representation Theorem.